

EXPRESS MAIL LABEL NO.
EL579666423US

5 METHOD AND APPARATUS FOR HANDLING A PLURALITY OF
TEST MODES
FOR A COMPUTER GENERATED DOCUMENT

10 Matthias Breuer
Andre Kuemmel

BACKGROUND OF THE INVENTION

15 Field of the Invention

The present invention relates generally to computer-based methods for testing scenarios and in particular to handling a plurality of alternative test configurations for a computer readable document like a text document, a spreadsheet document or the like.

Description of Related Art

Many computer programs allow the creation of documents containing a plurality of depending numbers for different, related configurations of the document. Examples of such programs are tax calculation programs with a fixed set of input and output options or spreadsheet programs, which form the basis for implementation of any kind of calculation application.

Fig. 1 schematically illustrates as an example of a computer-based spreadsheet document 100 having a plurality of parameters including rows 1, 2, 3, ... and columns A, B, C etc. In the example shown in Fig. 1, cells A1 and B1 are assigned parameter values 6 and 10, respectively. The contents of the cells C1, D1 and E1

are calculated based on the parameter values entered in cells A1 and B1. In the present example cell C1 has the value 3, cell D1 a value of 30, and cell E1 a value of 33.

5 If the user wishes to know the end result in cell E1 based on a different set of parameters values A, B, the user may, for example replace the value in cell A1 with 6, and replace the value in value in cell B1 with 7. If the user then wants to see the result for the
10 combination of parameters 6 and 12, the values of both cells A1 and B1 have to be changed and the calculation carried out accordingly

If such a program is used for carrying out complex calculations with a large number of related parameters,
15 it is likely that a user, at some stage, wants to "test" what effect some parameter changes have on the whole calculation. Such programs like tax calculation programs offer a "test" or "what if" mode for this purpose. The user activates the "what if" mode,
20 changes one or a plurality of input parameters, carries out the calculation based on these amended parameters and checks the result of the calculation.

If the user closes or exits the "what if" mode, the changed parameters are reset to their initial
25 values. If the user has activated the "what if" mode, it could happen that the user wants to test some different parameter values but ultimately wishes to return to the current parameter configuration. The user then has only two options. Either the user can
30 end the "what if" mode and start over from scratch, or the user can back up parameter by parameter until the user is back at the forking point. The larger the number of test paths, the more complicated the handling of different parameter sets becomes. It would
35 therefore be desirable to provide a more user-friendly handling of different test configurations of documents

having a large number of parameters dependent on other parameters.

Also, known in the prior art are version control systems. See for example, U.S. Patent No. 5,806,078.

5 However, such systems are not useful for testing the effects of various parameter sets in a computer-based method. In each case, the user would have to save a version at a point where a change in parameters was anticipated, complete the analysis, and then retrieve

10 the saved version and start over with the different parameters. This avoids the backing-up parameter by parameter. However, in a short period, the number of revisions and tracking the various versions within the version control system becomes an unwieldy task that

15 detracts from developing an understanding of the results from the testing. Also, if the user fails to save a version at the appropriate point, the user is simply back to the original task of backing up parameter by parameter. Moreover, such an approach

20 works only when sufficient storage is available for all the various versions that must be saved. Finally, if a user saves a number of versions that are not wanted, the user must manually delete the unwanted versions.

25 SUMMARY OF THE INVENTION

According to one embodiment of the present invention, there is provided a computer-implemented method of processing a computer generated document providing a user a plurality of nested test modes for

30 creating, reviewing and retrieving a plurality of different document configurations based on different document data sets. In one embodiment, the method includes:

activating, in response to a user request, a

35 first test mode of a document having a document data set;

automatically storing a first test mode data
 set of the first test mode where the first test
 mode data set is the document data set at the
 time of the activating the first test mode;
 5 changing data in the document data set in
 response to input of new data;
 activating, in response to another user
 request, a second test mode;
 automatically storing a second test mode data
 10 set of the second test mode wherein the second
 test mode data set is the document data set at the
 time of the activating the second test mode;
 changing data in the document data set in
 response to input of other new data;
 15 restoring the second test mode data set in
 the document upon leaving the second test mode;
 and
 restoring the first test mode data set in the
 document upon leaving the first test mode.
 20 Herein, a plurality of nested test modes means
 that a new test mode can be opened while at the same
 time an earlier test mode is still open. A sequence of
 multiple derived test modes can therefore be created.
 The user, thus, can activate a new test mode at any
 25 time even if the user is already in a test mode. It is
 therefore possible to return to any desired document.
 configuration. It is not necessary to know in advance,
 where a test mode should be started. The handling of
 complex documents with many different properties
 30 depending on the value of some parameters is therefore
 greatly facilitated.
 The document may be a spreadsheet document
 containing data arranged in a table, a text document,
 an Internet page, or any other type of document. The
 35 different configurations of the document may represent
 different formatting options of the document. It is

then easily possible to test a plurality of different formatting options of e.g. a publishing document, to initiate a test mode at any time a promising formatting configuration has been reached and to test further
 5 variations based on this configuration.

According to a particular embodiment, all different data configurations of the document within a test mode are stored and can subsequently be accessed by the user. The user can then go along a path of
 10 variations that the user has created within the test mode, for example, by operating forward and backward keys.

According to a particular embodiment of the invention the different test modes may be represented as a tree structure and displayed on a display screen.
 15 The user can then select a desired one of the displayed test modes. A specific tool for navigating to the displayed test modes may be provided. Additionally, it is possible to further gain access to different
 20 configurations within one test mode through the displayed structure of test modes.

According to a further specific embodiment the test modes and the corresponding data configurations are stored together with the document on a memory like
 25 a hard disk or a CD- or DVD-Rom. The user can then access the starting configurations of all test modes also when he opens a document at a future working session.

A further implementation of the present invention provides a computer system for processing a document providing a user with a plurality of nested or derived test modes for creating, reviewing and retrieving a plurality of different configurations of the document represented by different data sets, the computer system
 30 comprising a memory for storing the document data and a processing unit for carrying out opening a document,
 35

activating a first test mode upon user request, automatically, storing the starting document data set of the first test mode, changing the document data upon user input of amended data, activating a second test mode upon user request, automatically storing the starting document data set of the second test mode, changing the document data upon user input of amended data, restoring the starting document data set of the second test mode upon leaving the second test mode, and restoring the first document starting data set upon leaving the first test mode.

A still further implementation of the present invention may be realized by a computer program for processing a computer-readable document providing a user with a plurality of nested test modes for creating, reviewing and retrieving a plurality of different configurations of the document represented by different document data sets, the computer program comprising program code for activating a first test mode upon user request, automatically storing the starting document data set of the first test mode, changing the document data upon user input of amended data, activating a second test mode upon user request, automatically storing the starting document data set of the second test mode, changing the document data Upon User input of amended data, restoring the starting document data set of the second test mode upon leaving the second test mode, and restoring the first document starting data set Upon leaving the first test mode.

A program code may be embodied in any form of computer program product. A computer program product comprises a medium which stores or transports computer-readable code, or in which computer-readable code can be embedded. Some examples of computer program products are CD-ROM or DVD-ROM disks, ROM charts, floppy disks, magnetic tapes, computer hard drives,

servers on a network and signals transmitted over the network representing a computer-readable program code.

BRIEF DESCRIPTION OF THE DRAWINGS

5 Fig. 1 is an illustrated of a prior art computer generated spreadsheet.

 Fig. 2A is a block diagram of one computer system that includes an embodiment of the nested test mode module of this invention.

10 Fig. 2B is a block diagram of another computer system that includes an embodiment of the nested test mode module of this invention.

 Fig. 3 is a process flow for one embodiment of the nested test mode module of Figs. 2A and 2B.

15 Fig. 4 is a process flow for another embodiment of the nested test mode module of Figs. 2A and 2B.

 Fig. 5 schematically illustrates a tree representation of different test modes according to an embodiment of the present invention.

20 Fig. 6 schematically illustrates different formatting versions of a document representing different test modes according to an embodiment of the invention.

25 DETAILED DESCRIPTION

 In one embodiment of the present invention, a computer-based application 232 (Figs. 2A and 2B) includes a nested test mode module 230. Upon execution of application 232 including nested test mode module
30 230, a computer-generated document 295 is displayed on display unit 216. The user configures the computer generated document by entering a dataset , e.g., configures a spreadsheet as illustrated in Fig. 1, or perhaps provides display formatting directions. If the
35 user desires to test a different data set that includes one or more members, the use activates a test mode.

Upon activation of the test mode, nested test module 230 is activated. In one embodiment, module 230 stores the current data set on structure 220 as data set 0. In one embodiment, memory structure 220 is a memory stack, and in another embodiment is an associative memory array structure. See processes 300 and 400 that are described more completely below.

For the example of Fig. 1, data set 0 includes cell A1 with a value of 6 and cell B1 with a value of 10. The user then tests alternative sets of data by amending the data in the computer-generated document. For example, the user tries different values in cell B1 of Fig. 1.

If the user reaches a point where the user wants to try perturbations on a different set of data, the user again activates nested test module 230 that, in one embodiment, stores the current data set, i.e., the data set in the computer-generated document when the nested test module was activated, on memory structure 220 as data set 1. In the example of Figs. 2A and 2B, the user activates the nested test module 230 a third time and the current independent data at that time is saved as data set 2.

Hence, in this example, the user has nested three test modes. When the user closes the third test mode, data set 2 is removed from memory structure 220 and inserted in the document, and any dependent values are updated based on data set 2. The user can return to the original state by closing the second and first test modes, or alternatively, if the second test mode is the one the user wants to retain, the user simply saves the document after closing the third test mode, and then exits the document.

Here, use of a stack to store the data sets is illustrative only. In another embodiment, the data sets are stored in the associative array memory

structure where a test mode identifier is used as a key to access and restore a selected data set in a plurality of test mode data sets in the associative array memory structure. If desired the keys and
 5 corresponding data sets in the associative array memory structure can be exploded into a hierarchical tree structure on a permanent storage device. One important aspect is to store the data sets so the correct original document configuration can be restored upon
 10 the closing of a particular test mode.

With nested test module 230, the user tests various scenarios without having to worry about whether the document has been saved as a particular version as required with a version control system. Also, the user
 15 can easily restore the original data if the testing does not prove fruitful. Therefore, testing using complex documents is now easily performed by any user that is familiar with application 232.

As described above, when the user inputs a command to leave a test mode by pressing a corresponding key(s)
 20 or by clicking on a corresponding area on the screen in a graphical user interface, the system (i) checks which test mode is closed, for example the third test mode that saved data set 2, (ii) retrieves the corresponding starting data configuration, and (iii) changes all
 25 independent and dependent parameters of the document to the values based on this starting data configuration. When leaving a particular test mode, the user therefore has the same document with the same parameter values as
 30 when entering this particular test mode. If new dependent values have been created during the test mode, however, these new dependent values are not removed upon closing the test mode but updated based on the original independent parameter values.

35 In Fig. 2A, application 232 and module 230 are stored in a memory 284 of a server 280 and executed on

server 280 that is coupled to client computer system 200A by networks 203 and 204. The particular type of and configuration of client computer system 200A is not essential to this embodiment of the present invention.

- 5 The only requirement is that the client device be able to display the information described herein, and to issue input commands, as described herein. Similarly, networks 203 and 204 can be any network or network system that is of interest to a user that couples
- 10 client computer system 200A to server 280.

- Similarly, the method used to store and retrieve the data and/or data sets for the various test modes is illustrative only and is not intended to limit the invention to the particular embodiments described
- 15 herein. Finally, module 230 is illustrated as being a part of application 232 in Figs. 2A and 2B. Again, this is illustrative only. Module 230 must only be capable of being called from application 230.

- Fig. 2B is an example of when one embodiment of the present invention is implemented in a stand-alone computer system. The above comments with respect to Fig. 2A being illustrative only are directly applicable to Fig. 2B and are incorporated herein by reference.
- 20

- Fig. 3 is a process flow diagram for one embodiment of a method 300 that results upon executing of nested test module 230. The sequence of operations in method 300 is illustrative only and is not intended to limit the invention to the specific sequence shown. As is know to those of skill in the art, events can be
- 30 initiated at various points in time and when an event occurs, an event handler processes the event and branches to the appropriate location in method 300. For example, a new test mode can be created, or an existing test mode closed at any point.

- 35 After a user has created or opened a computer generated document in create document operation 301,

method 300 waits in test mode check operation 302 for a test mode activation event indicating that the user wishes to carry out a modification to data in the document, like the spreadsheet shown in Fig. 1, with
 5 some alternative values.

Typically, an application stays in create document operation 301 until the user initiates an input action other than data entry. This input action is passed to an event handler that in turn performs a test mode
 10 check operation 302. Typically, as is known to those of skill in the art, an event handler is used with a graphic user interface. Consequently, the return from test mode check operation 302 back to check operation 302 should not be interpreted as requiring processing
 15 to idle in check operation 302 until the event happens. Instead, it is one way of indicating that operations 303 to 308 are not initiated until a test mode activation event is received by method 300. Upon receiving the test mode activation event, processing
 20 transfers from check operation 302 to save data operation 303.

In save data operation 303, the data configuration of the document in which the user is presently working is stored as the starting data configuration of the
 25 test mode. As explained more completely below, when the test mode is closed completely, the system, in this embodiment, automatically returns this starting data configuration. Therefore, the user can always be sure that the user can return to this data configuration
 30 irrespective of the amendments the user makes while in the test mode.

Upon saving the data, operation 303 transfers to change data operation 304. In change data operation 304, the user makes any desired changes to the data
 35 configuration and initiates corresponding test calculations.

The user may come to a point where the user wants to save a second data configuration, while at the same time going on with testing of further alternative data sets. This is represented by new test mode check operation 305. If the user initiates another test mode activation event, new test mode check operation 305 transfers processing to save data operation 303.

Hence, the user activates a further nested test mode working in the same way as the first test mode.

10 Upon leaving the second test mode, the starting data configuration of the second test mode is restored. The user can, according to this embodiment, open as many nested test modes as the user wishes. For example, in the embodiments of Figs. 2A and 2B, the user opened 15 three nested test modes.

If the user does not initiate another test mode activation event, check operation 305 transfers processing to close test mode operation 306. If the user decides to leave the current test mode and 20 initiates a close test mode event, check operation 306 transfers to restore data operation 307 and otherwise to change data operation 304. The return to change data operation 304 should not be interpreted as requiring the user to make further changes to the data.

25 Rather, the current event is neither a new test mode event nor a close test mode event and so is not of interest for method 300. Upon completion of processing the current event, the user could issue immediately a close test mode event, or a new test mode event without 30 any further data changes, and so processing would simply pass through operation 304 in this case.

In restore data operation 307, the starting data configuration of the most recent test mode is restored in the document that the user has been testing.

35 Restore data operation 307 transfers processing to test mode open check operation 308. If the user has

additional test modes open, check operation 308 returns processing to new test mode check operation 305. If the user neither initiates a new test mode, nor closes another open test mode, the user is returned to change data operation 304.

When the test mode which has been opened first is closed, the starting data configuration of this first test mode is restored in operation 307 and check operation 308 transfers to end that does any necessary clean-up and exits process 300.

While in Fig. 3, operations 303 to 308 are illustrated sequentially, this is for ease of understanding only and is not intended to limit this embodiment of the present invention to sequential execution of the operations. As is known to those of skill in the art, parallel operations may be implemented with some processors and operating systems to achieve the results described above.

Moreover, check operations 305 and 306 should not be interpreted as requiring continuing polling for the particular events described. Typically, an application stays in change data operation 304 until the user initiates an input action other than data entry. This input action is passed to an event handler that in turn performs check operation 305 and 306, as necessary. The events associated with check operations 305 and 306 can be invoked at anytime after initiation of the test mode of operation, and are processed appropriately by the event handler.

If computer program 232 for processing the document the user is presently working on has a so-called "undo" function, which allows the immediate correction of the latest user action, one embodiment of the present invention may be implemented using this "undo" function as follows. The "undo" function creates a stack of "old" versions of the document.

Instead of storing the document data configuration upon initiating the test mode, the pointer to the corresponding data configuration of the document in the "undo", stack is saved in place of the data set. It is also possible to use versions of the document, which have already been saved in connection with the "undo" function, to save memory space.

In one embodiment of create document operation 301 and test mode check operation 302, when a test mode is activated in response to receipt of a user request, computer system 250A (Fig. 2A) and computer system 250B (Fig. 2B) first check in test mode check operation 302 which documents are presently opened and on which computer generated document (spreadsheet, text document, or others) the user is presently working. Check operation 302 next checks what parameters have been defined in this document, e.g. which cells of a spreadsheet table contain independent values. In the simple example of Fig. 1, only cells A1 and B1 contain independent parameter values and therefore have to be saved for the test mode. Note that when it is stated that a system takes some action, those of skill in the art understand that an instruction or instructions are executed that result in the stated action.

In this embodiment, save data operation 303 creates a data set containing all data necessary to completely define the starting data configuration of the document together with a test mode identifier, for example "test mode 1" or simple "1", as illustrated in Figs. 2A and 2B, which allows assigning the stored data configuration to a particular test mode.

Fig. 4 illustrates a further embodiment of the present invention, a process 400 that results from execution of another embodiment of nested test module 230. In process 400, operations with the same reference numeral as in Fig. 3 are equivalent to the

operations described with respect to Fig. 3, and so the description associated with Fig. 3 is incorporated herein by reference.

The sequence of operations in method 400 is illustrative only and is not intended to limit the invention to the specific sequence shown. Again, as is known to those of skill in the art, events can be initiated at various points in time and when an event occurs, an event handler processes the event and branches to the appropriate location in method 400. For example, a new test mode can be created, an existing test mode closed, or the test mode can be exited at any point.

Briefly, after creating or opening a computer generated document in create document 301 (Fig. 4), when a first test mode is activated upon user command, test mode check operation 302 transfers processing to save data operation 303 (Fig. 4).

As in the first embodiment of Fig. 3, the starting data configuration is subsequently stored in save data operation 303. In this embodiment, a test mode identifier is also stored with the data configuration. The test mode identifier is selected for recognizing and storing an order of creation of each different document data set associated with a different test mode. The node labels in Fig. 5 are one example of test mode identifiers.

In change data operation 304, the user enters parameter changes and carries out corresponding calculations, or tests other desired operations such as formatting the computer-generated document.

If the user initiates an event other than data input while in change data operation 304, the event handler performs test mode check operation 401. If the event is a test mode initiation event, check operation transfers processing to new test mode check operation

305 and otherwise to close test mode check operation
306. If the event was a close test mode event, close
test mode check operation 306 transfers to restore data
operation 307, and otherwise to exit test mode check
5 operation 405.

If the event was neither a test mode initiation
event, a close test mode event, nor an exit test mode
event, the event is not of interest to process 400 and
so processing returns to change data operation 304.

10 The return to change data operation 304 should not be
interpreted as requiring the user to make further
changes to the data. Rather, the current event is not
of interest for method 400. Upon completion of
processing the current event, the user could
15 immediately issue one of the three events without any
further data changes in operation 304.

If the event was a close test mode event,
operations 307 to 308 are performed as described above.
In one embodiment, the test mode event, the close test
20 mode event and the exit test mode event are initiated
upon selection of particular key combinations or
alternatively selection of particular items in a
graphic user interface.

In this embodiment, a test mode initiation event
25 can be either for a new test mode, or for a return to a
specific previous test mode. If the event is a new
test mode initiation event, check operation 305
transfers to save data operation 303 and otherwise
transfers to display test modes operation 402.

30 Display test modes operation 402 displays on a
display screen 216 the test modes, which have already
been created for the document on which the user
currently works. Fig. 5 is an illustration of one
embodiment of a display 500 in the form of a tree
35 structure. Each branching point of the tree structure
is assigned a test mode identifier that represents one

of the test mode data sets. Notice that two nests of test modes are displayed.

For complex documents like tax calculations or the like it may be useful to use a large number of nested test modes to carry out tax calculations for a large number of different parameter variations. For these cases, the display of the different test modes in a tree structure, as illustrated in Fig. 5, allows a user to get an overview of the different test modes.

The tree structure represents the parameter variations of the different test modes. Test mode 1 and test mode 2 correspond to different values of the same parameter. Test modes 1.1 and test mode 1.2 implement variations of a second parameter and test modes 1.2.1 and 1.2.2 different values of a third parameter etc. Thus, the test mode identifiers provide recognition of the order of creation of each different test mode parameter set. The test mode identifiers also provide an order for storing each of the different test mode parameter sets.

Alternatively, test mode 1 corresponds to a first set of parameters and test mode 2 a second different set of parameters. In this case, test modes 1.1 and 1.2 represent variations of a sub-set of parameters within the first set of parameters and so forth.

According to a particular embodiment of the present invention, a navigation tool is provided for the user to navigate, for example by movement of a mouse device or the like, between the different test modes in display 500. The navigation tool includes a unit for detecting the current mouse position, a unit directing the cursor following the mouse position, a unit for detecting the cursor position on the displayed tree structure of test modes. If the cursor position is on one of the test mode identifiers, an additional table may be displayed showing the parameter values

corresponding to this test mode. Similar navigation tools are used in many different types of computer programs.

- Using this displayed tree structure and a
- 5 corresponding navigation tool, when the user selects a particular one of the test modes of the document in display 500, select test mode check operation 403 transfers processing to restore data operation 404. Again, the user may not select a test mode for select
 - 10 test mode check operation 403 and instead issue a new test mode event, a close test mode event, or an exit test mode event. In each case, the event handler transfers processing to the appropriate check operation in method 400 and processing continues from that check
 - 15 operation.

- When the user does select a test mode from the displayed test modes, restore data operation 404 restores the parameter set associated with the selected test mode identifier in the document and transfers to
- 20 branch modes check operation 409. In contrast to the embodiment explained with reference to Fig. 3, it is therefore possible to change arbitrarily between all the test modes, which so far have been created.

- Branch modes check operation 409 determines
- 25 whether there are branch test modes from the test mode selected by the user. For example, if the user selects test mode 1.2, there are branch test modes 1.2.1, 1.2.2, and further branch modes 1.2.2.1 to 1.2.2.2 from test mode 1.2. In this case, check operation 409
 - 30 transfers to process branch modes operation 410 and otherwise to test mode check operation 401, if there are no branch test modes for the user selected test mode.

- In process branch modes operation 410, in one
- 35 embodiment, each of the branch test modes from the user selected test mode are marked as invalid. This

indicates that upon entry to the branch test mode, the test mode must be recalculated. In another embodiment of operation 410, each of the branch test modes is recalculated and saved. Operation 410 also transfers
 5 to test mode check operation 401.

In some cases, the user may decide that a particular set of test parameters is the only one of interest and all other test parameter sets may be discarded. In this case, the user wants to save the
 10 current test data as the valid data for the document and so the user issues an event to save only the current test mode.

In response to this event, exit test mode check operation 405 transfers to save all check operation 406
 15 that in turn transfers to save current test mode only operation 408. Save current test mode only operation 408 saves the document with the current test parameters as the actual document. In this case, all other test parameter sets are simply deleted, and method 400 ends.

In another situation, the user may want not only
 20 to stop, but also to return for further testing at another time based upon the current test configuration. In this case, the user wants to save all test data and so the user issues an event to save all the test modes.

In response to this event, exit test mode check operation 405 transfers to save all check operation 406
 25 that in turn transfers to save test modes operation 407. Save test modes operation 407 saves all the test modes with the document so that upon reopening the document, the test modes are available for further use.
 30 In this case, all test parameters sets are saved with the document, and method 400 ends. The embodiment of Fig. 4 is therefore particularly useful for complex documents where a large number of test modes involving
 35 many different parameters are created.

According to a further embodiment, it is possible to automatically store not only the starting data configuration but also every data configuration of the document the user creates within a test mode. The user
 5 can then track the test calculations the user has made in forward and in backward directions. According to a still further embodiment, it is possible to store the data configurations of the test modes together with the document in a permanent storage device.

10 The embodiments of the present invention are not only applicable to documents involving complex calculations like spreadsheet documents, but also the test modes, for example, may be applied also to different variations of a document. The user can try
 15 out different formatting options including character fonts, or perhaps various combinations of graphics, paragraphs etc., save promising versions as test modes or test documents and try out further improvements based on these test documents. Alternatively, the user
 20 can test graphic options such as borders, transparency, shadows, etc. For fonts, the user can test size, face, color etc. For paragraphs, the user can test borders, justification, line height etc.

In each case, the different test documents may be
 25 displayed as shown in Fig. 6. Test documents 601 to 603 are represented as tree structure including an area in which indications of the selected format like character font or paragraph layout are displayed. Alternatively, it is also possible to display a test
 30 page of the document when the cursor touches a document area. It is so possible for the user to easily review the different formatting options of the document.

The present invention is applicable to a hardware configuration 250B like a personal computer or a
 35 workstation as illustrated schematically in Fig. 2B. The computer may comprise a central processing unit

201, an input output (I/O) unit 202, an internal memory 210 and an external memory 211. Hardware configuration 250B may further include standard input devices like a keyboard 215, a mouse 218 or a speech processing means
 5 (not illustrated).

Another embodiment of this invention, however, may also be applied to a client-server configuration 250A as illustrated in Fig. 2A. The computer generated document may be displayed on a display screen of client
 10 device 200A while some or all operations of method 300 or alternatively method 400 are carried out on a server computer 380 accessible by client device 300 over a data network 203 and 204 as the Internet using a browser application or the like.

Herein, a computer program product comprises a medium configured to store or transport computer readable code for method 300, method 400 or both or in which computer readable code for method 300, method 400, or both are stored. Some examples of computer
 15 program products are CD-ROM discs, ROM cards, floppy discs, magnetic tapes, computer hard drives, servers on a network and signals transmitted over a network representing computer readable program code.

As illustrated in Figs. 2A and 2B, this storage
 25 medium may belong to the computer system itself. However, the storage medium also may be removed from the computer system. For example, method 300 may be stored in memory 284 that is physically located in a location different from processor 201. The only
 30 requirement is that processor 201 is coupled to the memory. This could be accomplished in a client-server system, or alternatively via a connection to another computer via modems and analog lines, or digital interfaces and a digital carrier line.

For example, memory 284 could be in a World Wide
 35 Web portal, while display unit 216 and processor 201

are in personal digital assistant (PDA), or a wireless telephone, for example. Conversely, the display unit and at least one of the input devices could be in a client computer, a wireless telephone, or a PDA, while
 5 the memory and processor are part of a server computer on a wide area network, a local area network, or the Internet.

More specifically, computer system 200A, in one embodiment, can be a portable computer, a workstation,
 10 a two-way pager, a cellular telephone, a digital wireless telephone, a personal digital assistant, a server computer, an Internet appliance, or any other device that includes the components shown and that can execute method 300, method 400, or both, or that at
 15 least can provide the display capability and input responses described herein. Similarly, in another embodiment, computer system 200A can be comprised of multiple different computers, wireless devices, cellular telephones, digital telephones, two-way
 20 pagers, or personal digital assistants, server computers, or any desired combination of these devices that are interconnected to perform, the methods as described herein.

Herein, a computer memory refers to a volatile
 25 memory, a non-volatile memory, or a combination of the two in any one of these devices. Similarly, a computer input unit and a display unit refer to the features providing the required functionality to input the information described herein, and to display the
 30 information described herein, respectively, in any one of the aforementioned or equivalent devices.

In view of this disclosure, methods 300 and 400 can be implemented in a wide variety of computer system configurations. In addition, methods 300 and 400 could
 35 be stored as different modules in memories of different devices. For example, method 300 could initially be

stored in a server computer 280, and then as necessary, a module of method 300 could be transferred to a client device 200A and executed on client device 200A. Consequently, part of method 300 would be executed on the server processor 282, and another part of method 300 would be executed on processor 201 of client device 200A. In view of this disclosure, those of skill in the art can implement the invention of a wide-variety of physical hardware configurations using an operating system and computer programming language of interest to the user.

In yet another embodiment, method 300 and/or method is stored in memory 284 of system 280. Stored method 300 and/or method 400 is transferred, over network 203, 204 to memory 211 in system 200A. In this embodiment, network interface 283 and I/O interface 202 would include analog modems, digital modems, or a network interface card. If modems are used, network 204 includes a communications network, and method 300 is downloaded via the communications network.

Methods 300 and 400 may be implemented in a computer program including comprehensive office application STAROFFICE that is available from Sun Microsystems, Inc. of Palo Alto, CA. (STAROFFICE is a trademark of Sun Microsystems.) Such a computer program may be stored on any common data carrier like, for example, a floppy disk or a compact disc (CD), as well as on any common computer system's storage facilities like hard disks. Therefore, an embodiment of the present invention also relates to a data carrier for storing a computer program for carrying out the inventive method. Another embodiment of the present invention relates to a method for using a computer system for carrying out the presented inventive method. Yet another embodiment of the present invention relates

to a computer system with a storage medium on which a computer program for carrying out the presented inventive method is stored.

- 5 While the invention has been shown with reference to particular embodiments thereof, it will be understood by those skilled in the art that various other changes in the form and details may be made therein without departing from the spirit and scope of the invention.

10

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2